

Entwicklung eines webbasierten Dokumentenarchivs zur Verwaltung und Veröffentlichung von Lehrmaterialien auf einem LAMP-System (Linux-Apache-mysql-PHP)

Diplomarbeit

im Fach Digitale Bibliothek

Studiengang Informationsmanagement

der

Fachhochschule Stuttgart –

Hochschule für Bibliotheks- und Informationswesen

David Prüm

Erstprüfer: Prof. Dr. Wolf-Fritz Riekert

Zweitprüfer: Prof. Dr. Wolfgang von Keitz

Bearbeitungszeitraum: 1. August bis 2. November 2000

Stuttgart, November 2000[

Kurzfassung

Die hier vorliegende Arbeit umfasst die Konzeption und prototypenhafte Entwicklung eines webbasierten Dokumentenarchivs zur Speicherung und Publikation von elektronischen Dokumenten, welche für die Lehre an der Hochschule für Bibliotheks- und Informationswesen in Stuttgart (HBI) relevant sind. Dieses Dokumentenarchiv ermöglicht die passwortgeschützte Speicherung und Aktualisierung der Dokumente über das Internet. Die Dokumente sind mit formalen und inhaltlichen Metadaten versehen. Der Zugang zu den Dokumenten ist, entweder über ein Suchformular oder durch sogenanntes Browsen in Übersichtsseiten möglich. Die Suche findet nur in den gespeicherten Metadaten statt. Eine Volltextindizierung der Dokumente selbst erfolgt nicht.

Schlagwörter: Dokumentenarchiv, Webdatenbank, mySQL, PHP

Abstract

This document describes the design and prototyping of a web based document archive, which contains electronic documents relevant to the purpose of studying at the University of applied Sciences Stuttgart, School of Library and Information Science (HBI). This archive provides password protected access to save and update documents via the World Wide Web. Every document is accomplished by a set of formal and content related meta data. It is possible to access those documents either by a web form or by browsing. You can only search the meta data but not the content of the documents.

Key words: document archive, web database, mySQL, PHP

Inhaltsverzeichnis

Kurzfassung.....	2
Abstract.....	2
Inhaltsverzeichnis	3
Abbildungsverzeichnis.....	5
Abkürzungsverzeichnis	6
1 Überblick	7
1.1 Status	7
1.1.1 Professorenhomepages	8
1.1.2 OPUS (Online Publikationsverband der Region Stuttgart)	8
1.1.3 Die Digitale Bibliothek der virtuellen HBI.....	8
1.1.4 Das Textarchiv des studentischen Servers machno online	9
1.2 Mein Lösungsvorschlag.....	9
1.3 Warum LAMP?	9
1.4 Planung	10
1.5 Probleme	10
1.6 Ausführung	11
2 Spezifikation der Anforderungen.....	12
2.1 Gewünschte Funktionalität	12
2.1.1 Speichern	12
2.1.2 Formales und inhaltlich Erschließen.....	12
2.1.3 Such- und abrufbar machen	13
2.2 Technische Voraussetzungen.....	13
2.2.1 Zur Entwicklung verwendete Software	13
2.2.2 Serverseitige Software	14
2.2.3 Clientseitige Software.....	14
2.2.4 Dadurch bedingte Einschränkungen	15
2.3 Benutzeroberfläche	15
2.3.1 Rolle des normalen Benutzers	15
2.3.2 Rolle des Bearbeiters	15
2.3.3 Rolle des Administrators	16
2.4 Sicherheit	16
2.4.1 Authentizität	16
2.4.2 Zugangsbeschränkung	16

3	Konzeption.....	18
3.1	Metadaten	18
3.1.1	Verwendete Metadaten	19
3.2	Datenbank	21
3.2.1	Metadaten	22
3.2.2	Daten.....	23
3.2.3	Zugriffsberechtigung	23
3.3	Dateisystem.....	24
3.4	Benutzermanagement	25
3.5	Sessionmanagement.....	25
4	Realisierter Prototyp	28
4.1	Benutzersicht	28
4.2	Bearbeitersicht	32
4.3	Administrationssicht.....	37
5	Resümee und Ausblick	41
5.1	Vorteile für die Dozenten	41
5.2	Vorteile für die Studierenden.....	41
5.3	Ausblick.....	42
	Literaturverzeichnis	43
	Online Ressourcen	43
	Printmedien	43
	Anhang	44
	Datenbanktabellen.....	44
	Quelltexte	46
	Erklärung.....	47

Abbildungsverzeichnis

Abbildung 1: Entity Relationship Diagram.....	22
Abbildung 2: Startseite DA.....	29
Abbildung 3: Dokumentenliste	30
Abbildung 4: Detailanzeige	31
Abbildung 5: Anmeldung.....	32
Abbildung 6: Dokumentenverwaltung.....	33
Abbildung 7: Neues Dokument oben.....	33
Abbildung 8: Neues Dokument unten.....	34
Abbildung 9: Neues Dokument gespeichert	35
Abbildung 10: Dokument ändern – Formular	36
Abbildung 11: Schlagwortverwaltung	37
Abbildung 12: Passwort ändern	37
Abbildung 13: Startseite Administration	38
Abbildung 14: Benutzerverwaltung	38
Abbildung 15: Verfasser ändern	39
Abbildung 16: Kategorienverwaltung.....	39
Abbildung 17: Dokumentenverwaltung (admin)	40

Abkürzungsverzeichnis

DA	Dokumentenarchiv
DB	Datenbank
DC	Dublin Core
HBI	Hochschule für Bibliotheks- und Informationswesen
HTML	Hypertext Markup Language
LAMP	Linux Apache mySQL PHP
MIME	Multipurpose Internet Mail Extension
MS	Microsoft
OPUS	Online Publikationsverbund der Region Stuttgart
PDF	Portable Document Format
PHP	PHP Hypertext Preprocessor
VHBI	Virtuelle HBI

1 Überblick

In den letzten Jahren ist es, nicht nur an unserer Hochschule, in Mode gekommen, Lehrmaterialien, insbesondere Vorlesungsskripte, in elektronischer Form über das Internet zur Verfügung zu stellen. Diese Dokumente liegen häufig als HTML-Dokumente vor, oft aber auch in anwendungsspezifischen Formaten, wie z.B. als Datei des weitverbreiteten Textverarbeitungsprogramms Microsoft Word oder auch als PDF-Datei. Zu diesem Zweck pflegen viele der Professoren¹ mehr oder weniger umfangreiche Webseiten. Dies beginnt mit einer einzelnen Startseite mit kurzen Informationen zur Person des Professors und einer Liste von Links auf Microsoft Word Dateien zum Download. Sie können aber auch aus einer umfangreichen und weitverzweigten Sammlung von Webseiten bestehen, die neben Einführungen zu bestimmten Lehrfächern und Dokumenten zum Download, auch Literaturlisten, Übungsaufgaben, den Stundenplan des Professors und Linklisten mit interessanten Internetquellen enthalten.

Diese ständig wachsende Zahl von mehr oder weniger frei zugänglichen Dokumenten bieten den Studenten die Möglichkeit, schnell auf umfangreiche Informationsmengen zuzugreifen. Gleichzeitig ist aber festzustellen, dass der Überblick über die verfügbaren Informationen verloren geht. Dies hat mehrere Gründe. Zum einen sind die Homepages der Professoren nicht zentral und einheitlich auf einem Webserver untergebracht sondern verteilt auf zwei Webserver an der Hochschule (<http://www.hbi-stuttgart.de> und <http://v.hbi-stuttgart.de>, dieser auch unter weiteren Namen) und auf verschiedene private Websites wie z.B. <http://www.payer.de>, <http://www.frank-thissen.de> und <http://www.bernhardhuetter.de>. Des weiteren bieten nicht alle Professoren Informationen online an, und jeder Professor strukturiert sein Angebot auf eine andere Art und Weise. Für dieses Problem möchte ich einen Lösungsvorschlag anbieten.

1.1 Status

Momentan existieren zwar schon einige Ansätze, die genau dieses Problem in Angriff nehmen, aber eine zufriedenstellende oder auch nur hinreichende Lösung wurde bislang nur punktuell erreicht. Im folgenden werde ich einige der Ansätze kurz umreißen.

¹ Der einfachen Lesbarkeit halber, verwende ich nur die maskuline Form (z.B. Professor, Student, Benutzer usw.), hierbei ist die weibliche Form selbstverständlich unausgesprochen stets mitgemeint.

1.1.1 Professorenhomepages

Die Speicherung der Dokumente auf den Homepages der Professoren hat einige Vorteile. Unter anderem sind sie sehr flexibel in der Gestaltung, schnell änderbar und damit immer so aktuell, wie der jeweilige Dozent dies möchte. Auf der anderen Seite stehen die schon beschriebenen Nachteile dieser unorganisierten und dezentralen Lösung.

1.1.2 OPUS (Online Publikationsverband der Region Stuttgart)

Die Universität bietet mit OPUS² den Service eines über das Internet zugänglichen Dokumentenservers an. Dieser bietet ein einheitliches Datenformat, nämlich PDF, eine dauerhafte Speicherung und eine Katalogisierung auf der Grundlage des Dublin Core Metadata Element Set. Auf den ersten Blick handelt es sich hierbei um die perfekte Lösung des Problems. Betrachtet man aber, wie viele Dokumente der HBI tatsächlich in OPUS gespeichert sind, erkennt man, dass es doch Nachteile geben muss. Die HBI nimmt schon seit über einem Jahr an OPUS teil und es sind insgesamt acht Dokumente gespeichert. Keines dieser Dokumente ist ein Vorlesungsskript und es ist lediglich eine Diplomarbeit enthalten. Das Problem der geringen Nutzung von OPUS könnte an der geringen Bekanntheit an der HBI oder auch an der Art der Dokumente liegen. Vorlesungsskripte und andere Lehrmaterialien sind veränderlich und schnelllebig, sie verändern sich nahezu jedes Semester, manche sogar noch häufiger. Es ist auch sinnvoll, veraltete, nicht mehr zeitgemäße Informationen zu entfernen, um den Studenten keine Fehlinformationen zu liefern. Diese Möglichkeit der Änderung bietet OPUS nicht, und ist damit keine geeignete Lösung für unser Problem.

1.1.3 Die Digitale Bibliothek der virtuellen HBI

Im Wintersemester 1998/1999 entstand im Rahmen des Wahlpflichtfaches Digitale Bibliothek die „Digitale Bibliothek der Virtuellen Hochschule für Bibliotheks- und Informationswesen“³. Der Ansatz dieser digitalen Bibliothek bestand darin, alle für die Lehre relevanten Dokumente zu sammeln und in einer, nach dem Fächerkatalog der HBI strukturierten, Übersicht anzubieten. Das Problem dieses Ansatzes war, dass die komplette Pflege der Dokumente von den Studenten dieses Wahlpflichtfaches durchgeführt wurde. Da im Rahmen dieses kurzen Wahlpflichtfaches auch keine Programme zur Automatisierung oder Datenbankbindung erstellt oder eingesetzt werden konnten, war diese digitale Bibliothek nach dem Abschluss des Wahlpflichtfaches zum Scheitern verurteilt. Da

² Siehe auch <http://elib.uni-stuttgart.de/opus/doku/about.html>

³ Siehe auch <http://v.hbi-stuttgart.de/Bibliothek/>

es den Verfassern der Dokumente nicht möglich ist, ihre Dokumente selbst einzubringen und zu aktualisieren, wurde das Angebot nicht weiter gepflegt und die Inhalte veralteten.

1.1.4 Das Textarchiv des studentischen Servers machno online

Auf studentischer Seite wurde das Problem der verteilten und schwer auffindbaren Lehrmaterialien ebenfalls frühzeitig erkannt und es wurde ein Lösungsversuch gemacht. Mit dem sogenannten „Textarchiv“⁴ wurde ein ähnlicher Ansatz verfolgt wie mit der Digitalen Bibliothek der VHBI. Das Textarchiv krankt ebenfalls an mangelnder Pflege und dem Fehlen der Automatisierung.

1.2 Mein Lösungsvorschlag

Anhand der letzten beiden vorgestellten Projekte lässt sich deutlich erkennen, dass ein Lösungsbedarf besteht. Alle bisherigen Ansätze kann man im Bezug auf die Lehrmaterialien an der HBI als gescheitert betrachten. Sie scheiterten entweder an der mangelnden Automatisierung und Beteiligung der Verfasser oder an ihrem Anspruch auf dauerhafte Speicherung und an mangelnder Flexibilität. Ein weiterer Aspekt ist, dass manche Dokumente nur für die Hochschulöffentlichkeit zur Verfügung stehen sollen und nicht im gesamten Internet zugreifbar sein dürfen. Alle vorgestellten Lösungen bieten keine Möglichkeiten zur Einschränkung des Nutzerkreises an⁵.

Ich möchte eine Webanwendung entwickeln, die in erster Linie die Verfasser der Materialien einbezieht und ihnen die Möglichkeit bietet, ihre Dokumente an einem zentralen Ort zu veröffentlichen. Diese Anwendung soll über eine Webschnittstelle verfügen, einfach zu bedienen sein, den Verfassern vollen Zugriff auf die von ihnen gespeicherten Dokumente bieten und es ermöglichen die Veröffentlichung einzelner Dokumente auf das Intranet der Hochschule zu begrenzen. Diese Anwendung soll keine perfekte Erschließung der Dokumente bieten und auch keine ewige Speicherung gewährleisten. Sie soll in erster Linie eine schnelle Lösung für das beschriebene Problem darstellen.

1.3 Warum LAMP?

Als Plattform für die Realisierung der Webanwendung wurde ein sogenanntes LAMP-System gewählt. LAMP steht für **L**inux **A**pache **M**ySQL **P**HP, eine Kombination aus Betriebssystem, Webserver, Datenbanksystem und Skriptsprache. Ein LAMP System

⁴ Siehe auch <http://machno.hbi-stuttgart.de/Textarchiv/>

⁵ Das Textarchiv hätte zwar diese Möglichkeit genutzt sie aber nicht und weist auch nicht darauf hin.

bietet hohe Leistung zu einem niedrigen Preis. Die gesamte, für ein LAMP-System benötigte Software, steht als Open Source, das heißt kostenlos im Quelltext, zur Verfügung. Linux erhält als Betriebssystem für Webanwendungen immer größere Bedeutung. Der Apache Webserver zählt zu den meistgenutzten Webservern. MySQL ist eine schnelle relationale Datenbank, die sich besonders für Webanwendungen eignet. PHP [Bakken, Schmid 2000] ist eine relativ neue Skriptsprache, die direkt im Quelltext der HTML-Seiten eingebettet wird. Sie eignet sich besonders zur Erzeugung von dynamischen Webseiten mit Datenbankbindung.

Die freie Verfügbarkeit der Software ermöglichte es mir auch, die Anwendungsumgebung auf meinem persönlichen PC zu installieren und dort zu entwickeln. Diesen Vorteil hätte ich mit kommerziellen Programmen nicht gehabt. Zu guter letzt steht diese Plattform an der Hochschule auf zwei Servern zur Verfügung. Zum Testen habe ich v.hbi-stuttgart.de gewählt, da dieser eigens zur Unterstützung der Lehre eingerichtet wurde, als Alternative wäre außerdem noch machno.hbi-stuttgart.de, der studentische Server, in Frage gekommen.

1.4 Planung

Zunächst gilt es die Anforderungen zu spezifizieren. Dies geschieht in detaillierter Form in Kapitel 2. Die Grundanforderungen lassen sich auf drei Kernfunktionen zusammenfassen: Speichern, Erschließen und zugreifbar machen. Gespeichert werden sollen zum einen die Dokumente selbst, und wenn dies nicht möglich ist, eine URL, die auf diese Dokumente verweist. Die Erschließung der Inhalte habe ich am Dublin Core Metadata Set angelehnt, dieses allerdings stark vereinfacht. Zugreifbar sollen die Dokumente auf möglichst einfachem Wege sein. Einerseits durch das Browsen durch hierarchische Schlagwortlisten und auf der anderen Seite über die direkte Suche über ein Suchformular.

1.5 Probleme

Während der Entwicklung bin ich auf mehrere Probleme gestoßen. Es bestehen zwei Möglichkeiten der Speicherung der Dokumentendateien auf dem Server. Sie können direkt im Dateisystem des Servers abgelegt werden. Dies wirft allerdings einige Sicherheitsprobleme auf.

Die andere Möglichkeit, welche vorzuziehen ist, ist die Speicherung direkt in der Datenbank. Diese Lösung erachte ich als die bessere. Vor allem ist dies ein guter Weg, um die Zugriffskontrolle zu realisieren, da jede Datei über ein Skript ausgegeben werden muss, und der Zugriff nicht über den Webserver allein erfolgen kann. Im Prototypen erfolgt die

Speicherung zwar noch im Dateisystem, für das fertige Produkt ist jedoch geplant, die Speicherung in der Datenbank vorzunehmen.

Ein weiteres Problem stellte die Authentifizierung dar. Es muss gewährleistet sein, dass nur dazu berechnigte Personen Dokumente speichern und bereits gespeicherte Dokumente löschen und ändern können. Dazu wird eine Benutzerverwaltung und ein Sitzungsmanagement benötigt. Zur Benutzerverwaltung kann man sich einerseits der Benutzerwaltung des Datenbankmanagement Systems, in diesem Fall MySQL, bedienen oder eine eigene Benutzerverwaltung realisieren. Die Verwendung der datenbankinternen Lösung hat den Vorteil, dass keine Zugangsdaten für die Datenbank in den Skript-Dateien gespeichert werden müssen, sondern die Anwender sich direkt bei der Datenbank anmelden. Der Nachteil dieser Lösung ist, dass sie das Sessionmanagement⁶ stark erschwert und nur wenige Informationen über die einzelnen Benutzer speichert. Deshalb müsste in jedem Fall eine zusätzliche eigene Benutzerverwaltung geführt werden, die weitere Informationen, wie den vollen Namen des Benutzers und seine e-Mail-Adresse, verwaltet. Aus diesen Gründen habe ich mich entschlossen die komplette Benutzerverwaltung selbst zu gestalten. Das Sessionmanagement ist ein weiterer Punkt, der Beachtung verdient. Hierzu habe ich eine Funktion geschrieben, die nach einer ersten Anmeldung mit einem Passwort weitere Zugriffe über eine Sessionidentifikationsnummer (im weiteren auch Session-ID genannt) ermöglicht. In diese Funktion sind auch einige Sicherheitsüberprüfungen und ein Zeitfaktor integriert, der die Session nach einer Zeitspanne von zehn Minuten, in der keine Anfrage des Benutzers erfolgte, ungültig macht.

1.6 Ausführung

Im 3. Kapitel wird die entwickelte Konzeption zur Umsetzung der Anforderungen aus Kapitel 2 geschildert. Parallel zur Entwicklung der hier dargestellten allgemeinen Konzeption habe ich an einem Prototypen gearbeitet, der einerseits zur Bestätigung der Realisierbarkeit dient und außerdem auch nach seiner Weiterentwicklung zum Einsatz kommen kann. Dieser Prototyp wird in Kapitel 4 beschrieben. Die Arbeit schließt mit einer Zusammenfassung der Ergebnisse und des erzielten Nutzens, sowie einem Ausblick auf weiterführende Aufgabenstellungen.

⁶ Eine Session ist eine Anwendersitzung, die sich im allgemeinen durch eine eindeutige Identifizierbarkeit des Anwenders oder der Sitzung kennzeichnet. Sie ermöglicht z.B., dass Anwender in einer Sitzung mehrere Webseiten anfordern können und sich nur einmal mit Namen und Passwort anmelden müssen. Die Anmeldedaten werden bei den Folgeseiten über die zu dieser Session auf dem Server gespeicherten Daten bereitgestellt.

2 Spezifikation der Anforderungen

2.1 Gewünschte Funktionalität

2.1.1 Speichern

Das Dokumentenarchiv⁷ soll den Zugriff auf Dokumente in digitaler Form über das Internet ermöglichen. Daher müssen zunächst die digitalen Dokumente gespeichert werden. Grundsätzlich sollte es für die Dateiformate und Größen keine Einschränkungen geben. Einschränkungen sind lediglich durch das Übertragungsmedium Internet bzw. Intranet gegeben. So sind Dateien von mehreren MByte Größe nur in einem Intranet mit relativ hoher Bandbreite in zumutbarer Zeit herauf- oder herunterzuladen.

Da es aus verschiedenen Gründen nicht immer sinnvoll oder möglich ist, die Originaldokumente zu speichern, sollen ebenfalls URLs erfasst werden.

2.1.2 Formales und inhaltlich Erschließen

Um diese Dokumente später auffindbar zu machen, müssen verschiedene formale und inhaltliche Informationen zu den einzelnen Dokumenten – sogenannte Metadaten – erfasst und gespeichert werden. Da es sich um ein eingegrenztes thematisches Gebiet handelt, nämlich Lehrmaterialien, beschränke ich mich auf einen relativ kleinen Satz von Metadaten.

Formal:

- Titel
- Verfasser. Es sollen bis zu 3 Verfasser angegeben werden können.
- Dokumententyp. Zum Beispiel Diplomarbeit, Skript zur Vorlesung o.ä.
- Zeit. Da es bei den hier betrachteten digitalen Dokumenten schwierig ist, ein Veröffentlichungsdatum auszumachen, wird hier am besten der Zeitpunkt der Speicherung im Dokumentenarchiv festgehalten.
- Ausgabebezeichnung, Auflage oder Versionsnummer. Um die Veränderungen und den Stand von veränderlichen Dokumenten verfolgen zu können, sollten die Bearbeiter einen beliebigen Versionsvermerk speichern können.

⁷ Künftig werde ich Dokumentenarchiv auch in abgekürzter Form als DA bezeichnen

Inhaltlich:

- Eine kurze inhaltliche Beschreibung oder ein Abstract
- Schlagwörter. Eine Einschränkung auf mindestens ein bis maximal vier Schlagwörter ist vorgesehen, um der Vergabe überflüssiger oder nicht genau zutreffender Schlagwörter vorzubeugen.

2.1.3 Such- und abrufbar machen

Die gespeicherten Materialien sollen den interessierten Studenten, im weiteren einfach „Benutzer“ genannt, einfach und intuitiv zugänglich sein. Daher sollte es zum einen ein Suchformular geben, das es ermöglicht nach den wichtigsten gespeicherten Metadaten zu suchen (Titel, Verfasser, Dokumententyp, Schlagwort und Stichwort aus Titel und Abstract). Zum anderen soll es möglich sein mittels Hyperlinks durch, nach verschiedenen Kriterien strukturierte, Übersichten zu navigieren, also zu browsen. Dies hat mehrere Vorteile:

1. Der Benutzer stößt bei seiner Suche auf Dokumente mit ähnlichem Inhalt, auf die er bei einer exakt formulierten Suche möglicherweise nicht gestoßen wäre.
2. Es ist für Suchmaschinen möglich, die gespeicherten Daten ihrerseits zu indexieren und somit eine weitere Zugangsmöglichkeit zu schaffen.

2.2 Technische Voraussetzungen**2.2.1 Zur Entwicklung verwendete Software**

HTML-Kit Version 1.0 (Build 290 Beta 10)⁸ – Dieses Programm diene zur Erstellung des PHP-Codes und besitzt einige nützliche Funktionen, von denen ich hauptsächlich das Syntax-Highlighting und die automatische FTP-Upload-Funktion genutzt habe.

phpMyAdmin Version 2.0.5⁹ - Dies ist eine in PHP geschriebene Webanwendung die eine graphische Oberfläche zur Manipulation von MySQL Datenbanken bereitstellt. Dieses Werkzeug habe ich zum Anlegen der dem Dokumentenarchiv zugrundeliegenden Datenbank verwendet.

Darüber hinaus habe ich verschiedene Webbrowser zur Überprüfung der Funktionsfähigkeit benutzt, wie zum Beispiel: **Netscape Communicator** in verschiedenen Versionen für

⁸ Siehe auch <http://www.chami.com/html-kit/>

⁹ Siehe auch <http://phpwizard.net/projects/phpMyAdmin/index.html>

Linux und MS Windows, **Opera** Version 4.02 für MS Windows 9x, **Internet Explorer** in verschiedenen Versionen und **Lynx**¹⁰ Version 2.8.9dev.9 für Linux.

2.2.2 Serverseitige Software

Das Betriebssystem des Servers ist Linux. Und zwar die Distribution SuSE¹¹ in der Version 6.4 mit einem Betriebssystemkern in der Version 2.2.16. Als Webserver wird Apache¹² in der Version 1.3.12 verwendet. Für den Webserver steht ein Modul zur Verfügung, welches die direkte Unterstützung von PHP-Skripten ermöglicht. Zu Beginn der Arbeit wurde ein Modul der PHP-Version 3.0.16 verwendet, zwischenzeitlich wurde aber ein neueres Modul in der Version 4.0.2 installiert.

2.2.3 Clientseitige Software

Das Dokumentenarchiv soll plattformunabhängig zugänglich sein, daher wird auf eine spezielle Clientsoftware verzichtet. Es soll ein relativ einfacher Webbrowser genügen um auf die Inhalte zuzugreifen. Netscape Navigator und MS Internet Explorer, Opera und sogar textbasierte Browser wie beispielsweise Lynx sollen als Client verwendbar sein.

Ich möchte ganz gezielt auf jegliche clientseitige Programmierung verzichten, da sie zum einen die Privatsphäre der Benutzer verletzen könnte und zum anderen nicht jedem Benutzer zur Verfügung steht, bzw. von bestimmten Nutzern absichtlich nicht genutzt wird. Zur Sessionverwaltung und zum Speichern von Benutzerdaten wäre es zum Beispiel möglich, Cookies¹³ zu verwenden. Viele Nutzer sehen Cookies aber als Eingriff in ihre Privatsphäre an und lassen sie deshalb nicht zu. Weiterhin haben viele Nutzer aus Sicherheitsgründen die Unterstützung von clientseitiger Programmierung auf Webseiten deaktiviert. Beispiele für solche Anwendungen sind Java, ActiveX, JavaScript, Jscript oder VBScript. Die clientseitigen Programme und Skripte ermöglichen es, Webseiten interaktiver und komfortabler zu gestalten. Sie werfen neben den oben genannten Problemen aber auch das Problem der Inkompatibilität und mangelnden Unterstützung durch die einzelnen Browser auf.

¹⁰ Siehe auch <http://lynx.browser.org/>

¹¹ Siehe auch <http://www.suse.de/>

¹² Siehe auch: <http://www.apache.org/>

¹³ Cookies sind kleine Datensätze, die von einem Webserver an den Browser übermittelt werden und von diesem als Text auf dem lokalen Rechner abgespeichert werden. Wird erneut eine Verbindung zu diesem Webserver hergestellt übermittelt der Browser automatisch diesen Datensatz an den Webserver. Damit kann das Nutzungsverhalten von Internetnutzern über einen längeren Zeitraum analysiert werden. Sie werden auch oft zur „automatischen Anmeldung“ verwendet.

2.2.4 Dadurch bedingte Einschränkungen

Die Upload-Funktion der heutigen Browser unterstützt meist nur das Speichern einzelner Dokumente. Dies bedingt, dass alle Dokumente die in diesem Archiv gespeichert werden sollen, als singuläre Datei existieren. HTML-Dateien mit eingebundenen Grafiken, Java-Applets, externen Style-Sheets oder JavaScript-Dateien können somit nicht erfasst werden. Es besteht zwar die Möglichkeit solche „Mehr-Dateien-Dokumente“ mittels eines Archivprogramms wie beispielsweise WinZip zu einer Datei zusammenzufassen, allerdings ist dann der direkte Zugriff auf das Dokument nicht mehr möglich. In diesem Fall müsste zuerst die gepackte Datei auf den lokalen Rechner geladen, gespeichert und dann entpackt werden, um danach erst mit dem entsprechenden Programm, in unserem Beispiel einem Webbrowser, lesbar zu sein. Auf jeden Fall sollte auch es auch die Möglichkeit geben, solche gepackten Dokumente abzulegen.

2.3 Benutzeroberfläche

Ein weiterer wichtiger Punkt ist die Benutzeroberfläche. Für die Akzeptanz bei den Benutzern ist dies vielleicht sogar der wichtigste Aspekt überhaupt. Es scheint mir sinnvoll, die Benutzeroberfläche entsprechend der verschiedenen Rollen der Benutzer in verschiedene Benutzeransichten zu unterteilen, so dass für jede Benutzergruppe nur die für sie verfügbaren Funktionen sichtbar sind.

2.3.1 Rolle des normalen Benutzers

Der normale Benutzer hat keine besondere Zugangsberechtigung und kann ohne eine spezielle Anmeldung auf die Inhalte zugreifen. Für ihn stehen eigentlich nur die Such- und die Browsingfunktion zur Verfügung. Man könnte bei Bedarf auch noch eine Feedbackfunktion ergänzen, die es dem Benutzer ermöglicht weitere Dokumente vorzuschlagen oder auch nicht mehr gültige URLs zu melden.

Es kann zwar notwendig sein, den Zugriff auf bestimmte Dokumente auf das hochschulinterne Intranet zu beschränken, allerdings kann der Benutzer darauf keinen Einfluss nehmen. Er wird lediglich darauf hingewiesen, dass das Dokument, welches er gerade angefordert hat, nur für Nutzer des Intranets verfügbar ist. Eine andere Möglichkeit wäre es, diese Dokumente für externe Nutzer überhaupt nicht anzuzeigen. Aber ich halte es für sinnvoller, zumindest das Vorhandensein dieser Dokumente nachzuweisen.

2.3.2 Rolle des Bearbeiters

Die Rolle des Bearbeiters umfasst nicht nur den eigentlichen Verfasser des Dokuments. Es kann sich dabei z.B. auch um einen Professor handeln, der für die Lehre relevante

Dokumente aus externen Quellen erfasst oder auch Studienarbeiten oder Referate seiner Studenten im Dokumentenarchiv aufnimmt. Der Bearbeiter ist in diesem Kontext eigentlich nur eine Person, die einen Benutzernamen und das zugehörige Passwort besitzt, und damit berechtigt ist, Dokumente im Archiv zu speichern.

Die Maske zur Erfassung der Metadaten muss sehr einfach und übersichtlich gestaltet sein, denn es darf nicht als Mühe oder Anstrengung empfunden werden dort etwas einzugeben. Die Nutzung dieses Dokumentenarchivs wird wahrscheinlich bestenfalls durch eine Empfehlung an die Professoren gefördert werden und selbst wenn diese das Archiv willkommen heißen, werden sie es nur nutzen, wenn es für sie keinen erheblichen Zusatzaufwand darstellt.

2.3.3 Rolle des Administrators

Der Administrator muss sämtliche Möglichkeiten des Bearbeiters und noch darüber hinausgehende Rechte besitzen. Er muss zunächst die Zugangsberechtigung für neue Bearbeiter vergeben können und gegebenenfalls auch diese Zugangsberechtigung wieder entfernen können. Er muss den vollen Zugriff auf alle von den einzelnen Bearbeitern gespeicherten Dokumente haben, um in Problemfällen Hilfestellung leisten zu können. Außerdem muss er die ersten und grundlegenden Anpassungen des Systems vornehmen. So zum Beispiel das Festlegen der Dokumententypen, oder auch das Hinzufügen von neuen oder das Modifizieren von vorhandenen Dokumententypen. Auch die inhaltlichen Kategorien, unter denen die Dokumente eingeordnet werden, muss der Administrator nach Rücksprache mit den Bearbeitern festlegen.

2.4 Sicherheit

2.4.1 Authentizität

Es soll sichergestellt werden, dass die enthaltenen Dokumente wirklich vom genannten Verfasser stammen, und dass diese Dokumente nicht verfälscht wurden. Da die Dokumente nicht in jedem Fall vom eigentlichen Verfasser im Archiv aufgenommen werden, muss zumindest sichergestellt sein, dass für jedes Dokument eine Person verantwortlich gemacht werden kann, und dass die Person und der Zeitpunkt der letzten Änderung gespeichert werden.

2.4.2 Zugangsbeschränkung

Vorlesungsskripte oder beispielhafte Prüfungsfragen, und gegebenenfalls auch Lösungen derselben, dürfen nicht unbedingt der breiten Öffentlichkeit zugänglich gemacht werden.

Unter anderem spielt hier auch das Urheberrecht eine Rolle, welches in besonderen Fällen lediglich ein eingeschränktes Vervielfältigungsrecht vorsieht.

Um diese Zugangsbeschränkung zu gewährleisten muss es möglich sein, für jedes einzelne Dokument eine Veröffentlichungsbeschränkung auf das hochschulinterne Intranet festzulegen.

3 Konzeption

3.1 Metadaten

Eine gute Grundlage für die zu erfassenden Metadaten stellt der Dublin Core Metadaten-satz dar [Dublin Core Metadata Initiative, 1999]. Dieser von bibliothekarischen Gremien entwickelte Metadatensatz ist speziell zur Beschreibung elektronischer Dokumente bestimmt.

[**Dublin Core Metadata Initiative** 1996] führt die folgenden 15 Elemente des Dublin Core auf:

1. Titel (DC.TITLE)
2. Verfasser oder Urheber (DC.CREATOR)
3. Thema und Stichwörter (DC.SUBJECT)
4. Inhaltliche Beschreibung (DC.DESCRPTION)
5. Verleger bzw. Herausgeber (DC.PUBLISHER)
6. Weitere beteiligte Personen und Körperschaften (DC.CONTRIBUTORS)
7. Datum (DC.DATE)
8. Ressourcenart (DC.TYPE)
9. Format (DC.FORMAT)
10. Ressourcen-Identifikation (DC.IDENTIFIER)
11. Quelle (DC.SOURCE)
12. Sprache (DC.LANGUAGE)
13. Beziehung zu anderen Ressourcen (DC.RELATION)
14. Räumliche und zeitliche Maßangaben (DC.COVERAGE)
15. Rechtliche Bedingungen (DC.RIGHTS)

Diese Vielzahl von Elementen gewährleistet zwar eine ziemlich vollständige Beschreibung der Dokumente, bringt aber Probleme mit sich.

3.1.1 Verwendete Metadaten

Von den 15 Dublin Core Metadaten-Elementen werden einige Elemente vom Bearbeiter¹⁴ erfasst und einige andere automatisch beim Speichern des Datensatzes.

Vom Bearbeiter werden erfasst:

1. **Der Titel** (DC.TITLE). Dieser wird in einem Datenbankfeld als String gespeichert, eventuell vorhandene Zusätze werden entweder angehängt oder nicht beachtet.
Die Artikel am Anfang des Titels werden mitsortiert. Da der Zugang nicht primär über eine Titelliste erfolgen soll, ist dies zu verschmerzen, insbesondere da von einer eingeschränkten Anzahl von Dokumenten ausgegangen wird.
2. **Der Verfasser** (DC.CREATOR). Die Verfasser werden, um eine zuverlässige Suche zu gewährleisten, in zwei Feldern gespeichert, zum einen der Nachname und zum anderen der Vorname. Einem Dokument sollen mindestens ein bis maximal 3 Verfasser, oder sonstige Beteiligte zugeordnet werden. Zur Vereinfachung werden hier auch Körperschaften erfasst.
3. **Schlagwörter** (DC.SUBJECT). Es sollen mindestens ein bis maximal vier Schlagwörter für jedes Dokument vergeben werden. Die beste Lösung wäre es, einen bestehenden Thesaurus nutzen zu können. Da mir aber kein Thesaurus zur Verfügung steht und der Aufwand den Thesaurus in die Datenbank zu integrieren und ihn zu verwalten erheblich wäre, habe ich diese Möglichkeit verworfen. Die Schlagwörter können von den Bearbeitern selbst vergeben und in vorgegebene Kategorien eingeordnet werden.
4. **Inhaltliche Beschreibung** (DC.DESCRPTION). Der Bearbeiter soll eine kurze inhaltliche Beschreibung oder einen Abstract zum Dokument eingeben. Bei Bedarf kann diese inhaltliche Beschreibung auch umfangreicher ausfallen.
5. **Datum** (DC.DATE). Das Dublin Core sieht speziell hier auch noch sogenannte Schemes vor, die das Format des Datums bezeichnen und zusätzlich sogenannte Qualifier, die z.B. festlegen, ob es sich um das Datum der ersten Speicherung oder der letzten Änderung usw. handelt. Die Zeit der Speicherung im Dokumentenarchiv wird automatisch festgehalten. Zusätzlich steht dem Bearbeiter noch Feld zur Verfügung, in dem er den Stand, die Version oder auch einen beliebigen Ausgabevermerk festhalten kann.

¹⁴ Wie in Kapitel 2.3.2 definiert.

6. **Ressourcenart** (DC.TYPE). Hier kann der Bearbeiter aus einer Liste vordefinierter Dokumentenarten auswählen. Beispiele sind: Skript zur Vorlesung, Handbuch, Programm (was eine Ausnahme darstellt da es kein Dokument im eigentlichen Sinne ist), Quellenverzeichnis/Linkliste, Einführung, Diplomarbeit, Hausarbeit usw.
7. **Quelle** (DC.SOURCE). Die Quelle kann optional angegeben werden. Als Quelle kommen eigentlich nur URLs in Frage, die entweder allein für sich oder als Alternative zu einer Datei das Dokument repräsentieren. Diese Interpretation des Elements SOURCE entspricht zwar nicht den Vorgaben des Dublin Core ist aber das naheliegendste Äquivalent.

Automatisch wird neben dem Datum nur noch das **Format** (DC.FORMAT) und die **Ressourcen-Identifikation** (DC.IDENTIFIER) gespeichert. Dies aber nur, wenn eine Datei gespeichert wird. Wenn lediglich eine URL erfasst wird, wird das Format außer Acht gelassen. Gespeichert wird der MIME-Typ¹⁵, der beim Upload einer Datei vom Browser übermittelt wird. Die Ressourcen-Identifikation ist gleichzusetzen mit dem Wert des Primärschlüssels der Dokumenten-Tabelle.

Nicht verwendet werden folgende Elemente:

1. **Verleger bzw. Herausgeber** (DC.PUBLISHER). Da die Werke meist direkt von den Verfassern veröffentlicht werden, kann auf dieses Element verzichtet werden.
2. **Weitere beteiligte Personen und Herausgeber** (DC.CONTRIBUTORS). Da es sich bei den zu erwartenden Dokumenten überwiegend um Textdokumente handelt werden die beteiligten Personen entweder als Verfasser genannt oder nicht aufgenommen.
3. **Sprache** (DC.LANGUAGE). Es ist davon auszugehen, dass vorwiegend deutschsprachige und in geringerem Maße auch englischsprachige Dokumente vorkommen. Meist wird die Sprache aus dem Titel ersichtlich sein, Ausnahmen sind vernachlässigbar.
4. **Beziehung zu anderen Ressourcen** (DC.RELATION). Dieses Element wird in [Dublin Core Metadata Initiative 1996] als noch experimentell bezeichnet und da wir hier primär von eigenständigen Dokumenten ausgehen wird dieses Element nicht benötigt.

¹⁵ MIME steht für Multipurpose Internet Mail Extensions, der MIME-Typ wurde ursprünglich verwandt, um Datentypen von Mailanhängen festzulegen.

5. **Räumliche und zeitliche Maßangabe** (DC.COVERAGE). Für dieses Element besteht im eingeschränkten Rahmen des Dokumentenarchivs keine Notwendigkeit.
6. **Rechtliche Bedingungen** (DC.RIGHTS). Auf die komplexe Frage der Rechte, insbesondere Urheberrechte, möchte ich hier nicht eingehen. Mit Ausnahme der Einschränkung des Benutzerkreises auf das Intranet sind hier keine weiteren Maßnahmen vorgesehen. Für Urheberrechts- und Verwendungsvermerke muss der jeweilige Verfasser im eigentlich Dokument selbst sorgen.

Zusätzlich zu den genannten Elementen werden noch zwei weitere Merkmale gespeichert. Zum einen die Identität des Bearbeiters¹⁶, um sicherzustellen, dass nur die Person, die das Dokument im Dokumentenarchiv gespeichert hat, es verändern und gegebenenfalls löschen kann, und um eine Kontaktperson bei Fragen zu diesem Dokument verfügbar zu haben. Zum anderen das Merkmal Intranet, welches die Beschränkung des Dokuments auf das Intranet kennzeichnet. Hierbei handelt es sich um ein Datenbankfeld, welches nur einen Wert aus einer vorher definierten Liste von möglichen Werten speichert. (Typ ENUM [Yarger et al. 2000 S. 246]). In diesem Fall können die Werte „JA“ und „NEIN“ gesetzt werden, wobei JA die Einschränkung in Kraft setzt und NEIN sie außer Kraft setzt.

3.2 Datenbank

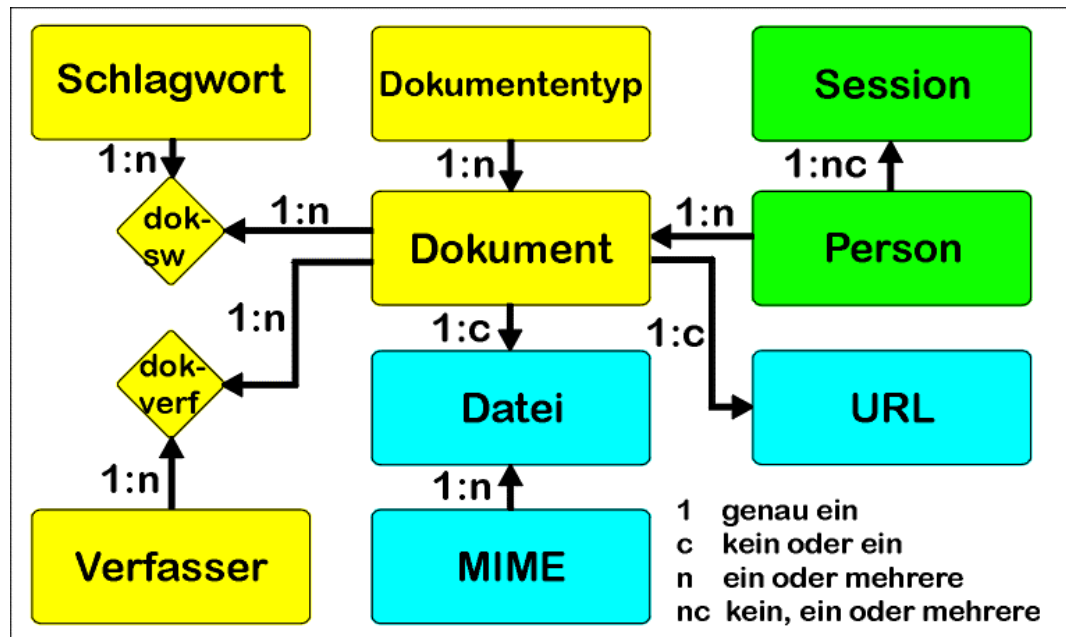
Zum besseren Verständnis habe ich im Text die Namen von Tabellen durch die Schriftauszeichnung **fett** kenntlich gemacht. Die Namen von Feldern sind unterstrichen.

Die Datenbank die dem Dokumentenarchiv zugrunde liegt hat drei Aufgaben:

1. Die Verwaltung der Metadaten (gelb dargestellt)
2. Die Verwaltung der Daten (blau dargestellt)
3. Die Verwaltung der Zugriffsberechtigung (grün dargestellt)

In *Abbildung 1: Entity Relationship Diagram* werden die logischen Verknüpfungen zwischen den einzelnen Tabellen des realisierten Prototypen dargestellt. In der Weiterentwicklung könnte die Tabelle **MIME**, die man strenggenommen auch zu den Metadaten zählen könnte, in dem die enthaltenen Werte direkt in der Tabelle **Datei** gespeichert werden.

¹⁶ Wie in Kapitel 2.3.2 definiert.

Abbildung 1: Entity Relationship Diagram¹⁷

3.2.1 Metadaten

Die Tabellen in denen die Metadaten gespeichert sind, sind in der *Abbildung 1: Entity Relationship Diagram* gelb eingefärbt. Sämtliche Metadaten die nur einfach vorkommen, also nicht wiederholt werden können, werden in der zentralen Tabelle **Dokument** gespeichert. Die einzige Ausnahme hiervon stellt der Dokumententyp dar. Da der Dokumententyp nicht frei vergeben werden darf, sondern aus einer Auswahlliste selektiert werden soll, stehen zwei Möglichkeiten zur Verfügung. Zum ersten könnte ein Datenfeld vom Typ ENUM verwendet werden. Dies würde bedeuten, dass eine Liste der möglichen Typen bei der Definition der Datenbank angegeben werden muss und weitere Typen nur durch eine Änderung der Datenbankdefinition hinzugefügt werden können. Zum zweiten kann die Liste der möglichen Dokumententypen in eine eigene Tabelle ausgelagert werden und neue oder veränderte Typen können problemlos in diese Tabelle eingefügt werden. Aus diesem Grund habe ich mich für die zweite Möglichkeit entschieden und die Dokumententypen in eine eigene Tabelle ausgelagert.

Die wiederholbaren Datenfelder Verfasser und Schlagwort stehen jeweils in einer n:m-Beziehung zum Metadatensatz und müssen daher beide über eine Zwischentabelle verknüpft in eigene Tabellen gespeichert werden.

¹⁷ Die Darstellungsweise ist angelehnt an [Riekert, 2000] Seite 11

3.2.2 Daten

Die Tabellen, in denen die Daten gespeichert sind, sind in der *Abbildung 1: Entity Relationship Diagram* blau eingefärbt. Als Daten betrachte ich die Dokumente die als Datei direkt im System gespeichert werden, sowie die URLs die stellvertretend für Dokumente stehen, die über das Internet zugreifbar sind.

Im augenblicklichen Stadium des Prototypen werden die Dateien noch im Dateisystem des Servers direkt gespeichert, daher ist es notwendig die MIME-Typen über vordefinierte Werte mit zugehörigen Dateiendungen und Beschreibungen in einer separaten Tabelle zu speichern. Wenn aber die Dokumente selbst ebenfalls in der Datenbank gespeichert werden, ist es ausreichend, den vom Browser des Bearbeiters mitgeteilten MIME-Typ in der Tabelle **Datei** direkt abzulegen. Die Trennung zwischen den beiden Tabellen **Dokument** und **Datei** habe ich vorgenommen, da ich später die Daten direkt in der Datenbank in der Tabelle Datei speichern will. Falls ich die für die Dateien benötigten Felder in die Tabelle **Dokument** aufnehme, würden diese Felder in vielen Datensätzen nicht gefüllt werden, was im Sinne eines ökonomischen Datenbankdesigns nicht wünschenswert ist. Ein weiterer Vorteil ist, dass durch das Hinzufügen einer Zwischentabelle eine n:m-Beziehung zwischen **Dokument** und **Datei** hergestellt werden kann.

Die URL ist aus den zuletzt genannten Gründen ebenfalls in eine eigene Tabelle ausgelagert. Diese Trennung zwischen Daten und Metadaten macht auch die beliebige Zuordnung von Datei oder URL, beziehungsweise Datei und URL zu einem Dokument möglich.

3.2.3 Zugriffsberechtigung

Die Tabellen in denen die Zugriffsberechtigung gespeichert und verwaltet wird, sind in der *Abbildung 1: Entity Relationship Diagram* grün eingefärbt. Die Zugriffsberechtigung für die Bearbeiter und den Administrator werden über die zwei Tabellen **Person** und **Session** verwaltet. In der Tabelle **Person** werden der Realname, das heißt der volle Vor- und Nachname, der Loginname, das Passwort in verschlüsselter Form und die e-Mail-Adresse der Bearbeiter gespeichert. Somit enthält sie folgende Felder: p_id (Primärschlüssel), name (Loginname, unique), pw (Passwort verschlüsselt) vorname, nachname, mail (e-Mail-Adresse). Mittels der p_id ist jedes Dokument genau einem Bearbeiter zugeordnet.

In der Tabelle **Session** wird bei der Anmeldung über den Webbrowser ein Sessiondatensatz mit einer eindeutigen Sessionidentifikationsnummer angelegt, nachdem das Passwort überprüft wurde. In diesem Datensatz werden die Personenidentifikationsnummer (p_id), die IP-Adresse und die Browserkennung, sowie die aktuelle Zeit gespeichert. Für jeden weiteren Seitenaufruf benötigt der Benutzer nun keine Anmeldung mehr sondern nur die

Sessionidentifikationsnummer. Um einem Missbrauch dieser Nummer vorzubeugen wird bei jedem Zugriff überprüft ob die IP-Adresse und die Browserkennung, die mit der Anfrage nach einer Seite übermittelt werden, mit den gespeicherten Daten übereinstimmen und ob das Zeitlimit nicht überschritten wurde. Nachdem dieser Vergleich positiv ausgefallen ist, wird die gespeicherte Zeit aktualisiert und die angeforderte Seite freigegeben. Andernfalls wird eine Fehlermeldung zusammen mit einer Aufforderung zum erneuten Login ausgegeben.

3.3 Dateisystem

Zu Beginn der Planung muss die Entscheidung stehen, wie und wo die gespeicherten Dateien vorgehalten werden sollen. Es gibt die Möglichkeit der Speicherung als normale Datei im Dateisystem, und die der direkten Speicherung in der Datenbank.

Das Abspeichern der Daten im Dateisystem des Servers wirft ein zentrales Problem auf, nämlich das der Zugriffsrechte. Aus Sicherheitsgründen läuft der Prozess des Webserver unter Linux mit den Rechten eines nichtprivilegierten Benutzers. Dieser hat nahezu keine Rechte auf das Dateisystem schreibend zuzugreifen. Dieser Benutzer kann nur auf die Dateien zugreifen, die explizit für alle freigegeben sind (world readable oder world writable), oder auf Dateien deren Eigentümer er ist. Die PHP-Skripte werden mit den Rechten dieses nichtprivilegierten Benutzers ausgeführt. Um nun das Abspeichern von Dateien durch diese PHP-Skripte zu ermöglichen gibt es zwei Wege:

1. Es wird ein spezielles Datenverzeichnis angelegt und diesem werden globale Schreibrechte zugewiesen (world writable).
2. Es wird ein spezielles Datenverzeichnis angelegt. Dieses wird dem selben Benutzer übereignet unter dessen Berechtigung auch der Webserver-Prozess läuft.

Beide Möglichkeiten bergen Probleme. Die erste Möglichkeit ist aus Sicherheitsgründen auf einem Mehrbenutzersystem, wie es Linux ist, nicht annehmbar, da so jeder beliebige Benutzer diese Dateien manipulieren oder löschen kann. Die zweite Möglichkeit steht nur dem Systemadministrator des Servers zur Verfügung. Diese Einschränkung kann zwar im Einzelfall akzeptiert werden, ist aber aus Gründen der leichten Installation auf anderen Systemen nicht wünschenswert. Außerdem kann in diesem Fall von jedem beliebigen Skript auf diesem Webserver auf diese Daten zugegriffen werden und die Daten können auch modifiziert oder gelöscht werden.

3.4 Benutzermanagement

Da die Möglichkeit Daten im DA abzulegen auf bestimmte, identifizierbare Personen begrenzt werden muss, ist es notwendig eine Art Benutzermanagement für das DA zu entwickeln. Hier bieten sich zwei Lösungsmöglichkeiten an. Zum einen kann die datenbankinterne Benutzerverwaltung verwendet werden. Dies bedeutet aber, dass sich jeder Benutzer mit dem Datenbankpasswort über das Internet anmelden muss, was zum einen als ein mögliches Sicherheitsrisiko angesehen werden kann und es zum anderen erforderlich macht, dass der DA-Administrator die Berechtigung besitzt, neue Datenbankbenutzer anzulegen. Weiterhin wirft diese Lösung das Problem auf, dass möglicherweise benötigte Zusatzinformationen, wie zum Beispiel die e-Mail-Adresse, separat gespeichert werden müssen, da die datenbankinterne Benutzertabelle keine Möglichkeiten vorsieht, weitere Informationen über die Benutzer zu erfassen. Der Vorteil der DB-internen¹⁸ Lösung wäre allerdings, dass keine vollprivilegierten Benutzerdaten im Skriptcode der Anwendung gespeichert werden müssen. Es ist allerdings möglich, durch eine entsprechende Vergabe der Dateirechte, einen lesenden Zugriff auf die Skriptdateien durch unbefugte Benutzer auf der Kommandozeilenebene zu unterbinden.

Die andere Lösung besteht darin die Benutzerdaten des DA in einer eigenen Tabelle zu speichern und zu verwalten. Dies hat den entscheidenden Vorteil, dass beliebige Zusatzinformationen abgelegt werden können. Weiterhin sind die Zugangsdaten wie Passwort und Benutzername völlig unabhängig von den systemintern verwendeten Daten, was einen weiteren Vorteil aus Sicht des Datenbankadministrators darstellt. Ich habe die Lösung der eigenen Benutzerverwaltung gewählt, um nicht in die Benutzerverwaltung der Datenbank eingreifen zu müssen, und weil ich zumindest die e-Mail-Adresse und den Realnamen der Benutzer speichern will.

3.5 Sessionmanagement

Das Sessionmanagement stellt eine zentrale Funktion dar, die entscheidend für die komfortable Benutzung des DA ist. Erst das Sessionmanagement ermöglicht dem Benutzer nach einmaliger Anmeldung das ungestörte Arbeiten. Seit der PHP-Version 4.0 wird von der Skriptsprache ein internes Sessionmanagement zur Verfügung gestellt, welches dem Entwickler eine sehr einfache Integrationsmöglichkeit bietet. Da ich aber zu Beginn meiner Arbeit noch auf die Vorgängerversion angewiesen war, musste ich zu diesem Zweck eine eigene Funktion entwickeln.

¹⁸ DB steht für Datenbank, in diesem Abschnitt ist damit das Datenbankmanagementsystem gemeint

```
function session(&$s_valid,&$SID,&$s_pw,&$s_user,
    $STD_DB, $STD_USER,$STD_PW,$SCRIPT_NAME,
    $HTTP_USER_AGENT,$REMOTE_ADDR,&$session_error)
{
```

Der Funktion werden alle Variablen übergeben die zur Verfügung stehen können und / oder die später zur Verfügung stehen sollen. Dazu werden bestimmte Variablen als Referenz angegeben. Dies wird durch das vorangestellte „&“ gekennzeichnet.

```
// Verbindung zur Datenbank wird hergestellt
mysql_connect("$STD_HOST","$STD_USER","$STD_PW");
mysql_select_db("$STD_DB");
$s_valid = 0; // Als Vorgabewert wird die Session
              // als ungültig betrachtet
// Eine Session ID ist vorhanden.
if($SID){
    $client_id = $HTTP_USER_AGENT;
    $client_ip = $REMOTE_ADDR;
    $s_query = "select client_id,client_ip,uid,
        pw,DATE_ADD(time, INTERVAL 10 MINUTE) as
        max_time,NOW() as akt_time from session where
        s_id=$SID";
    $s_result = mysql_query("$s_query");
    if($s_result && (mysql_num_rows($s_result) == 1)){
        $max_time=mysql_result($s_result,0,"max_time");
        $akt_time=mysql_result($s_result,0,"akt_time");
        $s_client_id= mysql_result($s_result,0,"client_id");
        $s_client_ip = mysql_result($s_result,0,"client_ip");
        if($max_time > $akt_time && $s_client_id == $client_id &&
            $s_client_ip == $client_ip)
        {
            $s_user = mysql_result($s_result,0,"uid");
            $s_pw = mysql_result($s_result,0,"pw");19
            $s_update = "update session set time=NOW() where s_id=$SID";
            $time_update = mysql_query($s_update);
            $s_valid = 1;
        }
        else {
            $session_error = "[HTML-Fehlermeldung]20";
        }
    }
    else {
        $session_error = "[HTML-Fehlermeldung]";
    }
}
// Es ist keine Session-ID vorhanden aber es sind
// Login-Informationen vorhanden
```

¹⁹ Das Passwort **\$s_pw** wird nur im Prototypen in der Session gespeichert. In der Endversion wird es nicht benötigt.

²⁰ An den so **[gekennzeichneten]** Stellen habe ich HTML-Passagen zur besseren Übersichtlichkeit gekürzt.

```

elseif($s_user && $s_pw){
    $s_ip = $REMOTE_ADDR;
    $client_sig = $HTTP_USER_AGENT;
    mt_srand(time());
    $S_ID = mt_rand(10000000, 99999999);
    $s_query = "select * from person where name=\"$s_user\"
                and pw = PASSWORD('$s_pw')";
    $s_result = mysql_query($s_query);
    if($s_result && mysql_num_rows($s_result) == 1){
        $s_insert = "insert into session values(\"$S_ID\",
            \"$client_sig\", NOW(), \"$s_ip\", \"$s_user\",
            \"$s_pw\")";
        if(mysql_query($s_insert)){
            $SID = $S_ID;
            $s_valid = 1;
        }
        else { $session_error = "[HTML-Fehlermeldung]; }
    }
    else {
        $session_error = "[HTML-Code für eine Passwortformular]";
    }
}
// Es sind weder Session-ID noch Login-Informationen
// vorhanden
else {
    $session_error = "[HTML-Code für eine Passwortformular]";
}
// Hier werden alle abgelaufenen Sessions gelöscht
$s_delete = "delete from session where
            DATE_ADD(time, INTERVAL 10 MINUTE) < NOW()";
$time_delete = mysql_query($s_delete);

```

Hier am Ende der Funktion werden Sessions mit abgelaufenem Zeitlimit gesucht und gelöscht. Somit wird die Sessiontabelle bei jedem Autorisierungsdurchlauf bereinigt.

```

mysql_close();
}

```

Die vorausgehende Funktion wird über eine inkludierte Datei jeweils am Anfang der geschützten Skriptseiten geladen und ausgeführt. Daraufhin wird anhand des Sessionstatus festgelegt, ob der Inhalt der Seite angezeigt wird oder die Fehlervariablen ausgegeben werden. In diesen Fehlervariablen ist, wie oben zu sehen, auch ein Formular zur Neuankmeldung enthalten.

4 Realisierter Prototyp

Der von mir entwickelte Prototyp erfüllt noch nicht alle zuvor gestellten Anforderungen, ist aber voll funktionsfähig und kann ohne tiefgreifende Veränderungen auf den gewünschten Stand gebracht werden. In diesem Prototyp werden die Dateien noch im Dateisystem abgelegt. Die Vergabe von neuen Schlagwörtern ist noch etwas umständlich und es fehlen Feedbackfunktionen, die Versions- und Ausgabebezeichnung sowie weitergehende Hilfeseiten. Den Hauptaufwand bei der Erweiterung dürften die hierarchisierten Schlagwörter, beziehungsweise inhaltliche Kategorien oder Themenkreise ausmachen, die aber bei der eingeschränkten Dokumentenzahl zu Beginn noch nicht benötigt werden. Die Kennzeichnung der Freigabe der Dokumente für das Internet oder nur für das Intranet ist ebenfalls noch nicht realisiert. Diese Funktion macht aber nur in Verbindung mit der Speicherung der Dateien in der Datenbank Sinn.

Der von mir erstellte Prototyp wird nun aus der Sicht der verschiedenen Anwender des Systems dargestellt. Das folgende Kapitel kann somit in gewisser Weise auch als Handbuch zum DA verwendet werden.

4.1 Benutzersicht

Der häufigste Zugriff auf das DA wird durch Benutzer erfolgen, die das System nicht genau kennen und lediglich schnellstmöglich ein bestimmtes Online-Dokument finden und darauf zugreifen wollen. Daher ist es wichtig, die Suche möglichst einfach und unkompliziert zu gestalten.



Abbildung 2: Startseite DA

Wenn ein Benutzer zum Beispiel ein Student die Adresse des DA aufruft (für den Prototypen <http://v.hbi-stuttgart.de/skriptarchiv/>) erscheint zunächst die in Abbildung 2 gezeigte Seite. In der Abbildung habe ich verschiedene Bereiche gekennzeichnet, um sie besser erläutern zu können.

Im Bereich 1 ist die Navigationsleiste zu sehen. Für einen Anwender ohne einen Benutzernamen oder ein Passwort sind nur die oberen drei Buttons²¹ in diesem Bereich von Bedeutung. Der erste Button führt ihn immer wieder auf diese Startseite zurück, so dass er, falls er eine neue Suche beginnen möchte, immer zu diesem Anfangspunkt zurückkehren kann. Der Button „Hilfe“ soll eine kurze Bedienungshilfe geben, ist aber im Prototypen noch nicht mit Inhalt hinterlegt. Der Button „Über DOKARC“ soll künftig auf eine Seite führen, die einige Informationen zur Anwendung selbst enthält.

Die Bereiche 2,3 und 4 bilden zusammen das eigentliche Inhaltsfenster. In diesem Bereich wird die jeweilige Suche formuliert und die Metadaten dargestellt. Hier kann natürlich auch das jeweilige Dokument ausgewählt werden. Im Bereich 1 kann ein freier Suchbegriff eingegeben werden, der nach dem Drücken des daneben befindlichen Buttons „Suche“ entweder eine Liste der gefundenen Dokumente, falls nur ein Dokument gefunden wird dessen Detailanzeige, oder die Meldung „Kein Treffer“ ausgibt. Der eingegebene

²¹ Button, englisch für Knopf oder Schalter, bezeichnet auf Webseiten einen Link zu einer weiteren Seite, welcher mit einer Grafik hinterlegt ist. Formulare besitzen ebenfalls solche „Schalter“, ihre Darstellung wird aber meist vom jeweiligen Browser bestimmt.

ne Suchbegriff wird im Titel, in den Schlagwörtern und im Abstract der gespeicherten Dokumente gesucht.

In Bereich 3 befindet sich eine Auswahlliste in der sämtliche Verfasser aufgeführt sind, deren Dokumente im DA erfasst sind. In Klammer wird die Anzahl der Dokumente angezeigt. Wenn in dieser Liste ein Verfasser ausgesucht und der daneben angeordnete Suchbutton angeklickt wird, erscheint eine Liste von Dokumenten, an deren Erstellung die ausgewählte Person als Verfasser beteiligt war.

In Bereich 4 befindet sich eine Auflistung aller Schlagwörter, die für ein Dokument im DA vergeben wurden, wieder mit einer Angabe der Anzahl der relevanten Dokumente. Jedes dieser Schlagworte stellt einen Link dar, der, wenn er aufgerufen wird, eine Liste der mit diesem Schlagwort verknüpften Dokumente ausgibt. Nehmen wir einmal an, der Student hat das Schlagwort „Informationsethik“ ausgewählt. Das Ergebnis wird in Abbildung 3 gezeigt.

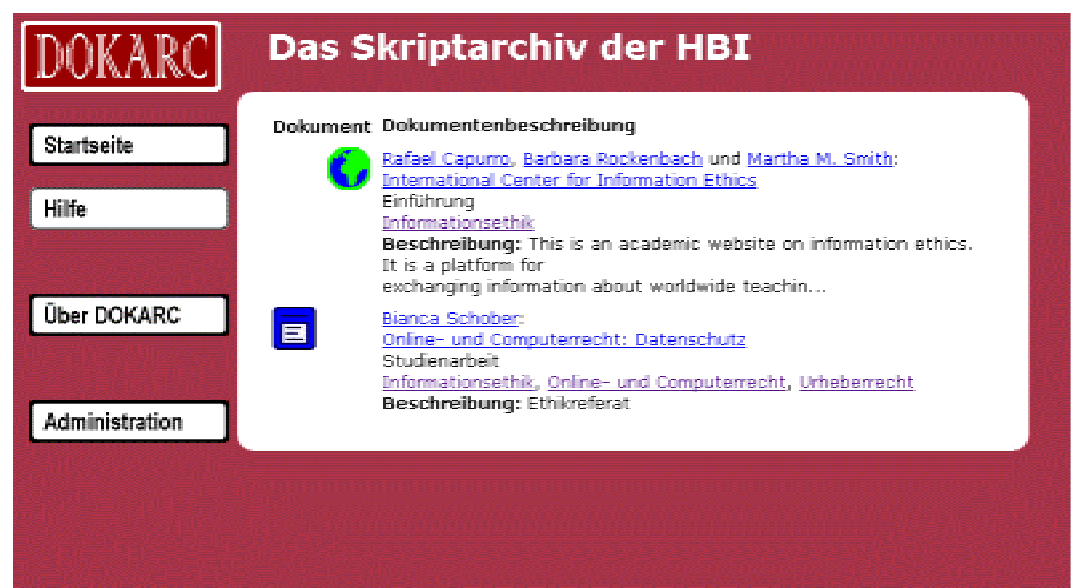




Abbildung 3: Dokumentenliste

In Abbildung 3 werden zwei Dokumente mit einer Kurzbeschreibung aufgelistet. Links von der jeweiligen Beschreibung befindet sich ein farbiges Symbol. Entweder , welches eine lokal im DA gespeicherte Datei repräsentiert oder , das einen Link zu einer externen Quelle repräsentiert. Es ist auch möglich, dass bei einem Dokument eine Datei und ein Link vorhanden sind. In diesem Fall werden beide Symbole nebeneinander dargestellt. Ein Klick auf eines dieser Symbole führt zum jeweiligen Dokument.

Rechts daneben werden in einer kurzen Beschreibung des Dokuments der oder die Autoren, der Titel, der Dokumententyp, die zugewiesenen Schlagworte und der Anfang der inhaltlichen Beschreibung bzw. des Abstracts angezeigt. Die Verfassernamen, der Titel und die Schlagworte sind jeweils als Link eingearbeitet. Ein Klick auf einen Verfassernamen führt zu einer Liste aller Dokumente dieses Verfassers, ein Klick auf ein Schlagwort führt zu einer Liste aller Dokumente, die mit diesem Schlagwort verknüpft sind und ein Klick auf den Titel des Dokuments führt zu einer Detailanzeige der Metadaten dieses Dokuments. Ein Klick auf den Titel „International Center for Information Ethics“ führt auf die in Abbildung 4 dargestellte Seite.

The screenshot shows a web interface with a dark red header and sidebar. The header contains the logo 'DOKARC' and the title 'Das Skriptarchiv der HBI'. The sidebar on the left has four buttons: 'Startseite', 'Hilfe', 'Über DOKARC', and 'Administration'. The main content area is white and displays the following information:

<p>Titel:</p> <p>Verfasser:</p> <p>Dokumenttyp:</p> <p>Schlagwörter:</p> <p>Dokument:</p> <p>Beschreibung:</p>	<p>International Center for Information Ethics</p> <p>Rafael Capurro, Barbara Rockenbach und Martha M. Smith</p> <p>Einführung</p> <p>Informationsethik</p> <p></p> <p>This is an academic website on information ethics. It is a platform for exchanging information about worldwide teaching and research in our field. It gives the opportunity to meet each other. It provides news on ongoing activities by different kinds of organizations. And it is free. The success of this website depends on the will of the people interested in this subject to share their knowledge with others.</p>
--	---

Abbildung 4: Detailanzeige

Zusätzlich zu den in der Listendarstellung angezeigten Informationen wird hier noch die vollständige inhaltliche Beschreibung angezeigt. In einer weiteren Entwicklungsstufe sollte hier noch angezeigt werden, wann und von welchem Bearbeiter dieser Metadaten-satz zuletzt gespeichert wurde und ob dieses Dokument nur für das Intranet freigegeben ist. Wiederum sind die Verfassernamen und das Schlagwort Links, die wie auf der Übersichtsliste auch zu neuen Dokumentenlisten führen. Diese Links in der Detailanzeige und in der Übersichtsliste dienen der schnellen und unkomplizierten Navigation zwischen den Dokumenten. Sie machen es sehr einfach, zum Beispiel ausgehend von einem Dokument,

alle Dokumente dieses Autors zu finden. Auch inhaltlich verwandte Dokumente lassen sich über die verlinkten Schlagworte leicht auffinden.

Die Bedienungsoberfläche für den Informationssuchenden ist gezielt einfach und übersichtlich gehalten, um eine möglichst intuitive Bedienung zu ermöglichen. Wer nach kurzem Probieren das Prinzip dieser Oberfläche verstanden hat, wird damit künftig kaum Probleme haben.

4.2 Bearbeitersicht

Als Bearbeiter muss man sich, bevor man verändernd auf seine Dokumente zugreifen oder neue Dokumente ablegen kann, zunächst anmelden. Um auf die Dokumentenverwaltung zuzugreifen, klickt man auf der Startseite auf den Button „Administration“ (Abbildung 2, Bereich 1), hierauf wird man zur Anmeldung aufgefordert (Abbildung 5).

Die Aufforderung zur Anmeldung wird solange wiederholt, bis man einen gültigen Benutzernamen und ein gültiges Passwort angibt oder auf den Button „Startseite“ drückt.

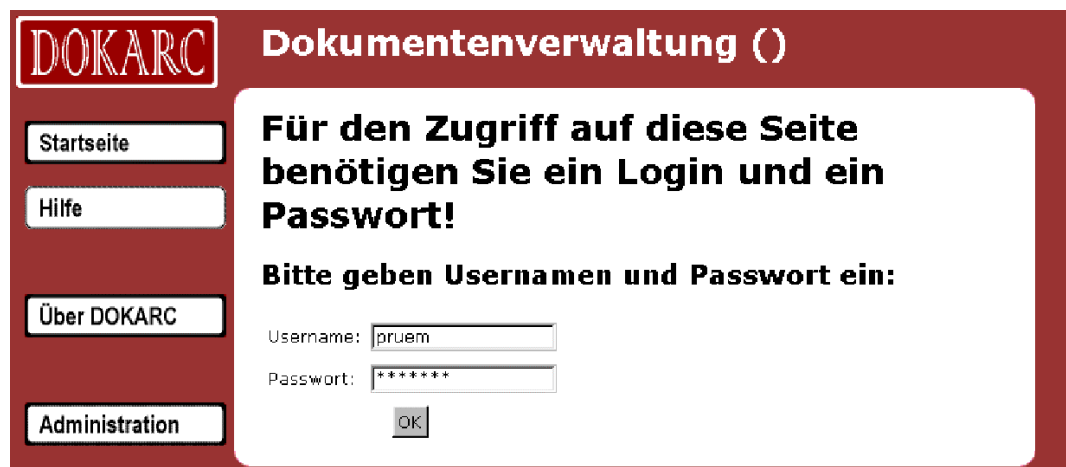


Abbildung 5: Anmeldung

Nach der erfolgreichen Anmeldung fällt zunächst auf, dass sich die Navigationsleiste verändert hat (Abbildung 6). Der erste Punkt ist nun „Benutzeransicht“. Dieser Button führt den Anwender zurück zur Startseite. Als neue Punkte sind hinzugekommen: „Neues Dokument“, „Dokument ändern“, „Schlagworte“ und „Passwort ändern“. Erhalten geblieben ist der Punkt „Hilfe“. Im Kopf der Seite ändert sich die Überschrift. Die neue Überschrift lautet „Dokumentenverwaltung“. Dahinter wird in Klammern der Benutzername unter dem der Bearbeiter sich angemeldet hat, angezeigt.

DOKARC **Dokumentenverwaltung (pruem)**

Benutzeransicht

Neues Dokument

Dokument ändern

Schlagworte

Passwort ändern

Hilfe

Eine kurze Anleitung zur Verwendung

Wenn Sie diesen Text lesen haben Sie sich mit einem gültigen Benutzernamen und Passwort angemeldet. Sie können nun folgende Funktionen nutzen:

- **Neues Dokument**
Mit dieser Funktion können Sie neue Dokumente ins Archiv aufnehmen. Sie können Dateien von Ihrem Rechner hochladen oder einfach eine URL angegeben auf die verwiesen werden soll. Wenn Sie möchten können sie auch beides angeben.
- **Dokument ändern**
Hier erhalten Sie eine Liste der von Ihnen gespeicherten Dokumente. Sie können eines davon auswählen und die gespeicherten Metadaten ändern oder den Datensatz löschen. Sie können auch die gespeicherten Dateien durch eine neue, aktualisierte Version austauschen.
- **Schlagworte**
Diese wichtige Funktion ermöglicht Ihnen das Hinzufügen von neuen Schlagworten und das Löschen von nicht benötigten Schlagworten. Wenn Sie beim Speichern eines neuen Dokumentes ein Schlagwort vermissen, lassen Sie es erstmal weg und speichern das Dokument.

Abbildung 6: Dokumentenverwaltung

Im weißen Inhaltsbereich der Seite wird nun zunächst eine kurze Bedienungsanleitung angezeigt. Der Button „Neues Dokument“ öffnet ein Formular zur Erstellung eines neuen Metadatensatzes und zur Speicherung einer Datei und / oder einer URL (Abbildung 7 und Abbildung 8).

DOKARC **Speichern eines neuen Dokuments (pruem)**

Benutzeransicht

Neues Dokument

Dokument ändern

Schlagworte

Passwort ändern

Hilfe

Titel:

Verfasser:

1.

2.

3.

Bitte verwenden sie die folgenden Eingabefelder für **neue Verfasser** nur, nur, wenn die gesuchten Verfasser nicht in den Dropdownlisten verfügbar sind. Bitte geben sie maximal 3 Verfasser an.

1. Neuer Verfasser: (Vorname) (Nachname)

2. Neuer Verfasser: (Vorname) (Nachname)

3. Neuer Verfasser: (Vorname) (Nachname)

Dokumententyp:

Abstract:

Literaturverzeichnis: Selbständige Quelle (Buch)

Abbildung 7: Neues Dokument oben

An oberster Stelle im Formular ist der Titel des Dokumentes einzugeben. Danach folgen die Verfasser. Bei den Verfassern ist zum einen zu beachten, dass maximal drei Verfasser angegeben werden dürfen, und zum anderen, dass man schon einmal gespeicherte Verfasser aus einer der Auswahllisten auswählen kann und noch nicht enthaltene Verfasser in den folgenden Formularfeldern eingeben kann. Sowohl das erneute Eintragen bereits existierender Verfasser, als auch die Eingabe von mehr als drei Urhebern sollte vermieden werden.

den werden. Jetzt folgt eine Auswahlliste mit vorgegebenen Dokumententypen, aus denen einer auszuwählen ist. Die Dokumententypen können nicht direkt eingegeben werden, um eine zu große Anzahl verschiedener Typen zu vermeiden. Nun folgt das Freitextfeld Abstract. Es dient nicht nur der Speicherung von Abstracts, sondern kann auch kurze, stichwortartige Beschreibungen oder andere inhaltliche Angaben aufnehmen (Abbildung 8, hier z.B. das Inhaltsverzeichnis).

Abbildung 8: Neues Dokument unten

Nun sind die Schlagworte zu vergeben. Es können bis zu vier verschiedene Schlagworte vergeben werden. Es muss mindestens ein Schlagwort vergeben werden, sonst wird das Dokument nicht gespeichert. Ganz am unteren Ende der Seite müssen nun die Angaben zum eigentlichen Dokument gemacht werden.

Entweder kann eine URL oder eine auf dem lokalen PC gespeicherte Datei angegeben werden. Für die Datei muss momentan noch der Dateityp manuell ausgewählt werden, in einer künftigen Version wird dies ebenfalls automatisiert ablaufen. Zur Auswahl der lokalen Datei bietet der jeweilige Webbrowser meistens einen Button an, durch den ein Fenster aufgerufen wird, welches die Suche im lokalen Dateisystem ermöglicht. Natürlich können auch eine URL und eine Datei angegeben werden, falls man zum Beispiel eine PowerPoint Präsentation einmal als Datei im programmeigenen ppt-Format und einmal im HTML-Format anbieten will. Es sollte in diesem Fall aber darauf geachtet werden, dass beide Dokumente den gleichen Inhalt haben und gleichzeitig aktualisiert werden.

Nachdem dieses Formular vollständig ausgefüllt ist, kann es mit einem Klick auf den „Speichern“-Button an den Server geschickt werden.

DOKARC **Speichern (pruem)**

Benutzeransicht

Neues Dokument

Dokument ändern

Schlagworte

Passwort ändern

Hilfe

Titel: Datenbankaufbau

Verfasser: Payer, Alois; Payer, Margarete

Dokumentenart: Skript zur Vorlesung

Schlagwörter: Datenbank, MySQL, Entity Relationship Model

Abstract: Grundkonzepte, Entwurf und Modellierung von Datenbanken, Entity Relationship Model, Relationale Datenbanken, SQL, Melvyl, dBase u.a.

Dokument: <http://www.payer.de/dblink.htm>

Titel: Datenbanken - Übungsklausur

Verfasser: Riekert, Wolf-Fritz

Dokumentenart: Skript zur Vorlesung

Schlagwörter: Datenbank, Entity Relationship Model

Abstract: Aufgabenblatt der Klausur vom 12. Juli 1999

Dokument: <http://v.hbi-stuttgart.de/skriptarchiv/daten/12.pdf>

Titel: Erstellung von wissenschaftlichen Arbeiten

Verfasser: Riekert, Wolf-Fritz

Abbildung 9: Neues Dokument gespeichert

Nachdem die Speicherung nun erfolgt ist wird eine Übersicht der gespeicherten Dokumente angezeigt. Falls das Formular nicht korrekt ausgefüllt wurde wird eine Liste der Fehler angezeigt und der Bearbeiter wird aufgefordert, den „Zurück“-Button seines Browsers zu betätigen um das Formular zu ergänzen. An dieser Stelle soll künftig eine Anzeige der gespeicherten Daten erfolgen.

DOKARC **Ändern (pruem)**

Benutzeransicht

Neues Dokument

Dokument ändern

Schlagworte

Passwort ändern

Hilfe

Willkommen David Prüm

In der folgenden Liste finden Sie die von Ihnen gespeicherten Dokumente. Sie können die Metadaten dieser Dokumente und die Dokumente selbst bearbeiten oder löschen.

Achtung! Wenn sie löschen anklicken wird das Dokument ohne Rückfrage gelöscht!

Bearbeiten	löschen	Ansehen
Datenbankaufbau	löschen	
Erstellung von wissenschaftlichen Arbeiten	löschen	
International Center for Information Ethics	löschen	
MySQL Datenbankhandbuch	löschen	
Tüpfli's Global Village Library	löschen	

Abbildung 10: Dokument ändern

Die nächste wichtige Funktion ist „Dokument ändern“. Hier erhält der Bearbeiter eine Liste aller Dokumente, die er gespeichert hat. Jedes Dokument wird mit seinem Titel aufgeführt. Danach folgt ein Link, der die Löschung des Dokuments bewirkt und zum Schluss ein Verweis auf das Dokument selbst, gekennzeichnet durch die schon erwähnte

Grafik. Durch das Anklicken des Dokumententitels wird ein Formular aufgerufen, welches das Ändern sämtlicher zuvor gemachter Angaben ermöglicht (Abbildung 11).

DOKARC **Ändern (pruem)**

Ändern des Dokuments: Erstellung von wissenschaftlichen Arbeiten

Titel:

Dokumententyp:

Verfasser 1: ☐ Verfasser 1 löschen

Neuer Verfasser:

Neuer Verfasser:

Bitte geben Sie insgesamt maximal 3 Verfasser an, überzählige Verfasser werden ignoriert. Um einen Verfasser durch einen anderen zu ersetzen löschen sie den ersten und geben den anderen als neuen Verfasser an.

Schlagwort 1: ☐ Schlagwort 1 löschen

Schlagworte:

Bitte geben Sie insgesamt maximal 4 Schlagworte an, überzählige Schlagworte werden ignoriert. Um ein Schlagwort durch ein anderes zu ersetzen löschen sie das erstere und geben das andere als neues Schlagwort an.

Abstract:

Benutzeransicht

Neues Dokument

Dokument ändern

Schlagworte

Passwort ändern

Hilfe

Abbildung 11: Dokument ändern – Formular

Prinzipiell ist dieses Formular mit dem Formular für neue Dokumente identisch, abgesehen davon, dass einige Werte vorgegeben sind. Bei den vorhandenen Verfassern und Schlagworten existiert dahinter jeweils ein Kästchen zum Ankreuzen. Wenn man dieses markiert, wird der vorstehende Verfasser aus diesem Dokument entfernt. Gleiches gilt für die Schlagworte. Alle anderen Daten, mit Ausnahme der Datei, können durch Auswahl eines anderen Listeneintrags oder Ändern des vorgegebenen Textes verändert werden. Um eine neue lokale Datei einzuspeichern, existieren wieder die gleichen Formularfelder, wie im oben genannten Formular für neue Dokumente. Hier gibt es noch ein zusätzliches Kästchen zum Ankreuzen mit dem eine gespeicherte Datei gelöscht werden kann, ohne sie durch eine neue zu ersetzen.

Hinter dem Punkt „Schlagworte“ verbirgt sich die Schlagwortverwaltung. Sie bietet eine Übersichtsliste aller vorhandenen Schlagworte und ermöglicht es, beliebige neue Schlagworte in die Liste der Schlagworte aufzunehmen. Es ist ebenfalls möglich, Schlagworte zu löschen. Allerdings nur solche, die in keinem Metadatensatz verwendet werden (Abbildung 12).

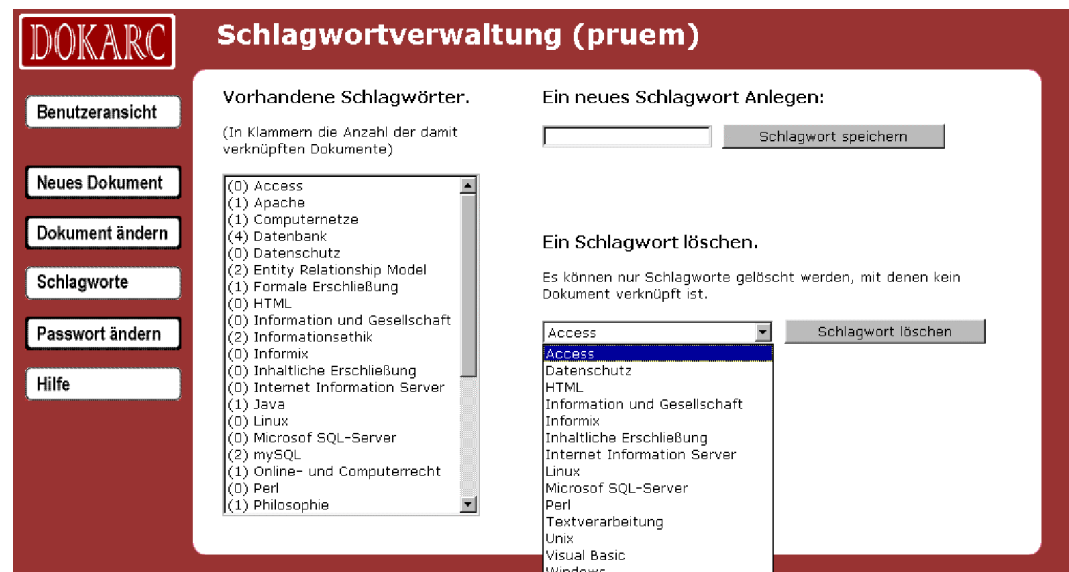


Abbildung 12: Schlagwortverwaltung

Als letzte Option besteht für den Bearbeiter noch die Möglichkeit sein Passwort zu ändern (Abbildung 13). Aus Sicherheitsgründen muss zunächst das alte Passwort eingegeben werden und danach zwei mal das neue.

Der Button „Hilfe“ führt zurück auf die schon beschriebene Anleitungssseite (Abbildung 6).

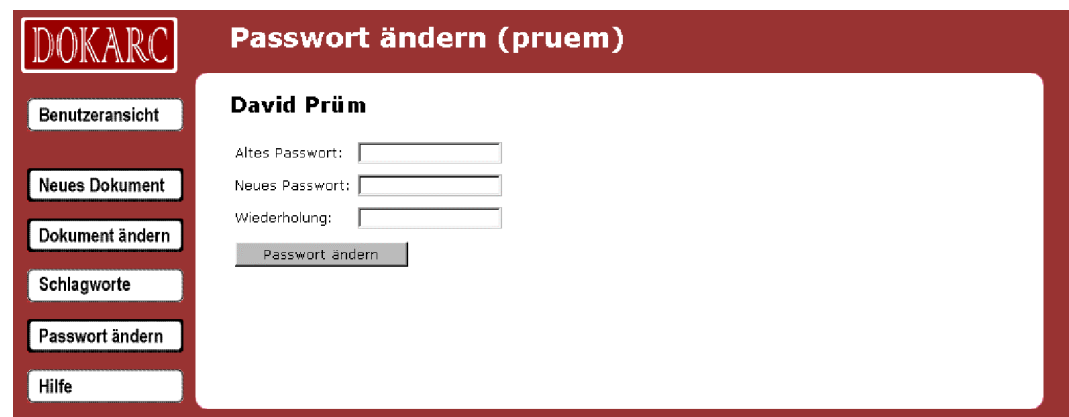


Abbildung 13: Passwort ändern

4.3 Administrationssicht

Der Administrator des DA muss die volle Kontrolle über alle Dokumente besitzen. Daher stehen ihm unter anderem alle Bearbeiter-Funktionen zur Verfügung. Darüber hinaus muss er neue Bearbeiter anlegen, vergessene Passworte ändern und alte Bearbeiter entfernen können.

Die Anmeldung erfolgt auf dem selben Weg wie die Anmeldung der Bearbeiter. Lediglich durch den Benutzernamen wird er von den anderen Bearbeitern unterschieden. Sobald er sich als „admin“ angemeldet, hat erhält er die selbe Oberfläche wie die anderen auch, lediglich die Navigationsleiste wird um einige Funktionen erweitert (Abbildung 14).

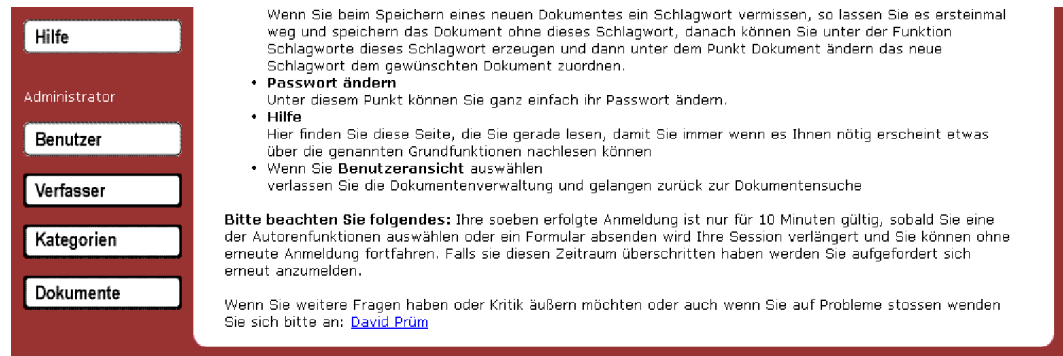


Abbildung 14: Startseite Administration

Die zusätzlichen Schaltflächen sind: „Benutzer“, „Verfasser“, „Kategorien“ und „Dokumente“. Nach Auswahl der Schaltfläche „Benutzer“ erscheint Abbildung 15.



Abbildung 15: Benutzerverwaltung

Hier stehen dem Administrator sämtliche Möglichkeiten der Benutzerverwaltung zur Verfügung. Er kann die Daten, inklusive Passwort, vorhandener Nutzer ändern, er kann

Benutzer löschen und komplett neue Benutzer anlegen. Als Zusatzoption kann er sich noch alle Dokumenten eines Nutzers anzeigen lassen.

Abbildung 16: Verfasser ändern

Die Schaltfläche „Verfasser“ bietet die Option vorhandene Verfassernamen zu ändern oder vorhandene Verfasser, sofern keine Dokumente von ihnen mehr gespeichert sind, zu löschen. Abbildung 16 zeigt das Formular zur Änderung eines Verfassernamens, der rechteckig umrahmte Bereich ist die Standardausgabe dieses Skriptes.

Abbildung 17: Kategorienverwaltung

Abbildung 17 zeigt die Kategorienverwaltung, die über die Schaltfläche „Kategorien“ aufgerufen wird. Hier können neuen Dokumententypen erzeugt, Dokumententypen geändert und gelöscht sowie Schlagwörter geändert und gelöscht werden.

Die letzte Administrationsfunktion ist die Dokumentenverwaltung, welche über den Button „Dokumente“ erreichbar ist. Diese Seite erlaubt es, Dokumente nach verschiedenen Kriterien zu selektieren und die selektierten Dokumente entweder einem anderen Bear-

beiter zuzuweisen oder sie zu löschen. Abbildung 18 zeigt das Ergebnis einer Suche nach den Dokumenten eines Verfassers

DOKARC

Dokumentenverwaltung (admin)

Benutzeransicht

Neues Dokument

Dokument ändern

Schlagworte

Passwort ändern

Hilfe

Administrator

Benutzer

Verfasser

Kategorien

Ergebnis der Suchen nach dem Verfasser mit der ID: 9

International Center for Information Ethics	
Dokumententyp:	Einführung
Verfasser:	Rafael Capurro; Martha M. Smith; Barbara Rockenbach;
Schlagworte:	Informationsethik,
Abstract:	This is an academic website on information ethics. It is a platform for exchanging information about worldwide teaching and research in our field. It gives the opportunity to meet each other. It p...
Eigentümer:	<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px 5px;">David Prüm (pruem)</div> <div style="margin: 0 5px;">▼</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">Eigentümer ändern</div> </div>
Löschen:	<div style="border: 1px solid #ccc; padding: 2px 10px;">Dokument löschen</div>

(1) Apache ▼

(1) Capurro, Rafael ▼

(5) David Prüm ▼

Suche nach Schlagwort

Suche nach Verfasser

Suche nach Eigentümer

Abbildung 18: Dokumentenverwaltung (admin)

5 Resümee und Ausblick

In der vorliegenden Arbeit wurden eine Konzeption und ein Prototyp eines webbasierten Dokumentenarchivs entwickelt. Dieses DA ist speziell auf die Bedürfnisse der Lernenden und Lehrenden der HBI zugeschnitten. Hiermit ist es möglich beliebige elektronische Dokumente über das Internet zu speichern und zu aktualisieren. Die Bedienung erfolgt durch übersichtliche Formulare auf serverseitig generierten Webseiten. Das DA bietet eine Benutzerverwaltung mit der für beliebige Bearbeiter ein schreibender Zugang geschaffen werden kann. Auch ein einfaches aber effektives Sessionmanagement ist integriert. Für die Informationssuchenden ist eine einfache Suchoberfläche gegeben, die den Zugang zu den Dokumenten über die Schlagworte oder Autoren sowie über eine Stichwortsuche bietet. Es gibt einen gesonderten Administrationszugang der es ermöglicht Dokumententypen und Schlagwörter in inhaltliche Kategorien zu gruppieren²². Die Speicherung der Daten erfolgt in einer Datenbank mit SQL-Schnittstelle. So können alle gespeicherten Daten auch für andere Anwendungen genutzt werden. Das DA bietet für Studenten wie Professoren einige Vorteile gegenüber der jetzigen Verfahrensweise.

5.1 Vorteile für die Dozenten

Wenn sich die Nutzung des DA durchsetzt, können die Dozenten an einer zentralen Stelle einen Überblick über alle von ihnen veröffentlichten Dokumente einsehen und diese Dokumente einfach bearbeiten. Sie können diese Dokumente nach ihrem Gutdünken inhaltlich erschließen und sie so in einen Kontext mit inhaltlich ähnlich gelagerten Dokumenten ihrer Kollegen setzen. Sie können sich unnötige Arbeit ersparen, wenn sie in bestimmten Fächern auf Dokumente ihrer Kollegen verweisen, anstatt dass sie für jedes Thema eigene Skripte erstellen. Sie erhalten alle Dokumente zu einem Thema auf einen Blick in einer Übersicht und können so ihre Lehrinhalte mit ihren Kollegen abstimmen. Die Dozenten können weiterhin auch externe Informationsquellen aus dem Internet in das DA integrieren und so, durch ihre Expertise gesichert, das Informationsangebot durch qualitativ hochwertige Inhalte ergänzen.

5.2 Vorteile für die Studierenden

Die Studierenden erhalten durch das DA einen schnellen und umfassenden Überblick über alle an der Hochschule verfügbaren elektronischen Dokumente. Sie erhalten eine

²² Die inhaltlichen Kategorien sind im Prototyp noch nicht realisiert.

schnelle und einfach zu bedienende Suchoberfläche, die ihnen den schnellen Zugang zu elektronischen Skripten gewährleistet. Sie haben, durch die namentliche Nennung des bearbeitenden Dozenten²³, eine Gewähr für die inhaltliche Qualität der dort vorgefundenen Dokumente.

5.3 Ausblick

Das DA in seiner jetzigen Form könnte als Ausgangspunkt für ein integriertes Informationssystem für die Lehre dienen. Eine denkbare Zusatzfunktion des DA könnten Dokumentensammlungen oder auch Verbunddokumente sein. Diese wären, speziell zu einer Lehrveranstaltung oder einem Modul zusammengestellte Dokumentensammlungen, die, ergänzt durch konventionelle Literaturangaben, den gesamten Lehrinhalt einer Veranstaltung abdecken könnten. Die Verbunddokumente würden von den jeweiligen Dozenten für eine Veranstaltung zusammengestellt.

Da auf dem Lehrservers v.hbi-stuttgart.de mit dem Stundenplansystem und dem Raumbelegungsplan jetzt schon einige andere Angebote auf der selben Datenbank-Basis existieren, wäre es relativ leicht realisierbar diese Komponenten so zu integrieren, dass zu jedem Eintrag im Stundenplansystem ein Link auf das zugehörige Verbunddokument vorhanden ist. Im Verbunddokument kann dann wiederum auf aktuelle Veranstaltungstermine hingewiesen werden.

In einem weiteren Schritt könnten personalisierte oder personalisierbare Informationsseiten für jeden Studenten erzeugt werden, die seinen persönlichen Stundenplan und zu jeder Veranstaltung auch gleich die benötigte Literatur anzeigen. Durch eine möglichst vollständige Vernetzung mit allen verfügbaren Informationsquellen, wie z.B. auch Adressverzeichnisse der Professoren, deren Sprechstunden und einem Terminkalender, könnte so eine weltweit über das Internet erreichbare, integrierte Arbeitsoberfläche für die Studenten und Dozenten entstehen.

²³ Gemeint ist der Bearbeiter nicht der Verfasser.

Literaturverzeichnis

Online Ressourcen

[Dublin Core Metadata Initiative 1996] Dublin Core Metadata Initiative: Metadata-Tags zur Erschließung von Internetquellen: Metadata-Elemente des Dublin Core. Max-Planck-Institut für Bildungsforschung, Berlin
URL: <http://www.mpib-berlin.mpg.de/DOK/metatagd.htm>
Letzter Zugriff am: 11.10.2000

[Dublin Core Metadata Initiative, 1999] Dublin Core Metadata Initiative: The Dublin Core Element Set Version 1.1
URL: <http://purl.org/DC/documents/rec-dces-19990702.htm>
Letzter Zugriff am : 11.10.2000

[Riekert 2000] Riekert, Wolf-Fritz: Datenbanksysteme und Aufbau von Datenbanksystemen. HBI Stuttgart
URL: <http://v.hbi-stuttgart.de/~riekert/lehre/db.pdf>
Letzter Zugriff am: 13.10.2000

[Bakken, Schmid 2000] Bakken, Stig und Schmid, Egon (Hrg.): PHP Hypertext Preprocessor. PHP Documentation Team
Letzter Zugriff am: 29.10.2000

Printmedien

[Yarger et al. 2000] Yarger, Randy Jay; George Reese und Tim King: MySQL & mSQL : Eine Datenbank für kleine und mittelgroße Unternehmen & Web Sites. O'Reilly, Köln ISBN 3-89721-163-7

Anhang

Datenbanktabellen

Tabelle Art:

Feld	Typ
art_id	int(11)
dok_art	varchar(160)

Tabelle Datei:

Feld	Typ
d_id	int(10)
name	varchar(30)
m_id	int(11)
d_time	datetime
dok_id	int(11)

Tabelle dok_sw:

Feld	Typ
dok_id	int(11)
sw_id	int(11)

Tabelle dok_verfasser:

Feld	Typ
dok_id	int(11)
v_id	int(11)

Tabelle Dokument:

Feld	Typ
dok_id	int(11)
titel	varchar(255)
abstract	text
zeit	datetime
stand	varchar(100)
person	int(11)
art_id	int(11)
intranet	enum('JA','NEIN')

Tabelle Mime:

Feld	Typ
m_id	int(11)
descr	varchar(30)
type	varchar(30)
suffix	varchar(6)

Tabelle Person:

Feld	Typ
p_id	int(11)
name	varchar(50)
pw	varchar(40)
vorname	varchar(100)
nachname	varchar(100)
mail	varchar(60)

Tabelle Schlagwort:

Feld	Typ
sw_id	int(11)
p_id	int(11)
schlagwort	varchar(160)
zeit	timestamp(14)

Tabelle Session:

Feld	Typ
s_id	bigint(20)
client_id	varchar(120)
time	datetime
client_ip	varchar(15)
uid	varchar(20)

Tabelle URL:

Feld	Typ
dok_id	int(11)
url	varchar(254)

Tabelle Verfasser:

Feld	Typ
v_id	int(11)
nachname	varchar(60)
vorname	varchar(60)
time	timestamp(14)
p_id	int(11)

Quelltexte

Die Quelltexte zum Dokumentenarchiv, sowie einige enthaltene Beispieldokumente sind neben den SQL-Datendefinitionen auf dem beigefügten Datenträger enthalten.

Nähere Informationen zum Inhalt entnehmen Sie bitte der Datei `readme.txt` auf dem Datenträger.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig angefertigt habe.
Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt.
Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ort, Datum

Unterschrift